

IPMI/IPMB Satellite Controller Firmware Update Procedure

On the pages that follow, you will find the firmware update procedure for the IPMI/IPMB Satellite Controller used in CompactPCI power supplies. The firmware was developed by Philips Semiconductors using their P89C662HBA/00 controller. Detailed information on this controller, including features, complete specifications, application notes, parametrics, support and tools, can be found at <http://www.semiconductors.philips.com/pip/p89c662hba/00>.

Philips Semiconductors

IPMI/IPMB Satellite Controller Firmware Update Procedure

*James P. Earle
Market Applications Engineer
Philips Semiconductors
2140 Lake Park Blvd., Suite 200
Richardson, Texas 75230
972-705-2485
jim.earle@philips.com*

IPMI/IPMB Satellite Controller Firmware Update Procedure

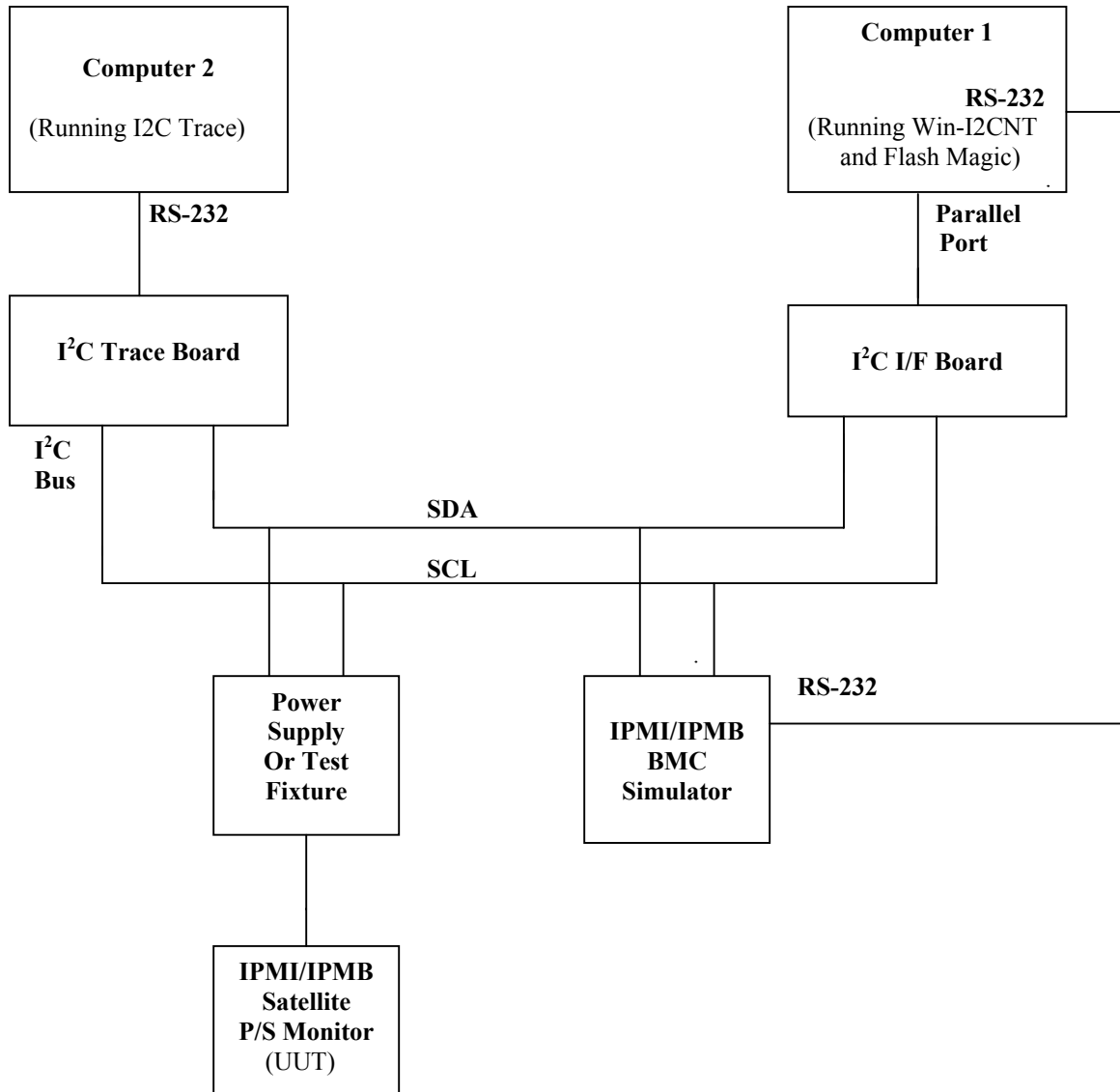


Figure 1. Test Set up

Test Description

The purpose of this test procedure is to functionally test and evaluate the IPMI/IPMB interface for a power supply in an IPMI system. The IPMI protocol is a request / response protocol. The IPMI/IPMB satellite controller being tested will typically communicate with the baseboard master controller since it manages all of the sensor data records and event logs. IPMI/IPMB protocol provides for other devices to also be able to interrogate satellite controllers, event data logs, etc. However, the test described here, will only simulate communication with the baseboard master controller (BMC). Two computers are not required for this test. However, using two computers facilitates the testing.

Set-Up description:

The test set-up consists of the following: There are two computers, an I²C trace board with an RS-232 interface, an I²C interface board with a parallel port interface, a BMC simulator (with both I²C and RS-232 interfaces), the IPMI/IPMB satellite controller unit under test (UUT), a power supply or test fixture to connect to the UUT, and the necessary cables and power supplies to interconnect the various pieces.

The BMC Simulator for this test is a Phytex 89C66x demo board that has a female DB-9 connector (P1) for an RS-232 interface to a computer, and a male DB-9 connector (P2) for interfacing to the I²C bus. The jumper settings on the Phytex board are as follows:

JP2: 3-4	(boot switch connects to Vcc)
JP3: 2-3	(+5V operation)
JP8: 1-2	(RxD and TxD connected to P1 DB-9)
3-4	
JP9: 1-2	(GND Connected to P2 DB-9)
3-4	(I ² C to P2 DB-9)
JP12: 1-2	(I ² C to P2 DB-9)

The I²C signals on the P2 connector are wired as follows:

SDA:	Pin 1
SCL:	Pin 4
GND:	Pin 5

Software programs used include “WinI2CNT[®]” and “I2C Trace” from The Boardshop (<http://www.demoboard.com>), and “Flash Magic” from the Embedded Systems Academy (www.esacademy.com). Flash Magic is used to download Intel hex files to the Phytex demo board via the RS-232 bus.

Computer #1, running WinI2CNT along with an I²C parallel port interface adapter, simulates the BMC issuing commands to the UUT. The UUT then responds to the command that was embedded in the request issued by the BMC. The I²C protocol requires that for each byte transmitted, an acknowledge bit will be transmitted back to the transmitter by the receiver (by holding the SDA line low during the acknowledge bit time). Computer #1 with the I²C interface board, does not have the capability to provide

the acknowledge signal back to the transmitter. The BMC simulator board provides the acknowledge function of the BMC. There is also the case of the satellite controller issuing an event message. An event message is generated if one of the sensors scanned by the satellite controller exceeds an assertion threshold, or comes back within the deassertion threshold, assuming the proper enables are set. This is the only case where the satellite controller issues the request. The BMC simulator also functions to provide a response to event messages generated by the satellite controller. The BSIM, for this test, also functions as a memory buffer and interface. This allows a hex file that is downloaded to the BSIM via the RS-232 bus to be downloaded to the UUT via the I²C bus. As a result the entire request/ response process and firmware update process can be tested under realistic conditions. Computer #2, running I2C Trace software (with an I²C trace board connected to the serial port), allows monitoring of the bus so both the requests and the responses can be observed and compared with the expected responses detailed in the specification.

The procedure categories and the associated test files are listed below:

- Firmware Update Commands (/c_firmware)
 - fw_test.mem
- BSIM Control g_bsim_cont
 - bsim_cont2.mem

Procedures

1. Set-up

Configure the test equipment and UUT as shown in Figure 1. The tests described here assume the BMC Simulator is a Phytex 89C66x demo board loaded with “bmc_update.hex” firmware. Devices with this software are marked B-SIM2.

Computer #2 should be running I2C Trace from The Board Shop (demoboard.com), or similar I2C trace utility. It is assumed computer #1 is running Win-I2CNT from The Board Shop. Choose “Device”, then “Universal” from the pull down menus. The files referred to in the following instructions can be loaded from the “File” pull-down menu.

2. Downloading Hex files from the computer to the Phytex demo board via RS-232

On the Win-I2CNT software, select Device – Universal, then File – load, and select the following test file: g_bsim_cont/bsim_cont2.mem. The following commands can be issued to the BSIM (Phytex demo board):

- 1- Erase Boot Vector / Status Byte
- 2- Program Boot Vector (FCh)
- 3- Program Status Byte (FFh)
- 4- Cold Reset (of the BSIM)
- 5- Download (used for BSIM – UUT download)

Procedure:

Step through the first 4 commands listed above. This will put the BSIM in the “RS-232 boot loader” mode. In this mode Flash Magic is used to communicate with the BSIM. Using Flash Magic perform the following:

- 1- Fill in the set-up parameters
- 2- Check – Erase block 0 and block 1
- 3- Load the appropriate hex file (i.e., IPMI_FLSH1.hex)
- 4- Check “Verify after programming”
- 5- Hit “Start”

This downloads the hex file to the BSIM. To place the BSIM into the “I²C boot download” mode perform the following using Flash Magic:

- 1 – Select “ISP” pull-down menu, then “Boot Vector and Status Byte”.
- 2 – Change the Boot Vector to 41, click OK and answer yes.
- 3 - Select “ISP” pull-down menu, then “Reset”

This places the BSIM back into its normal operating mode with a hex file for updating stored in its memory. The following firmware update can now be performed.

3. Firmware Update

Open the file “c_firmware_up/fw_test.mem”. The requests generated are listed as follows:

- 1 – Start F/W Update Mode
- 2 – Erase Blocks 0 and 1 (takes 12-13 seconds)
- 3 – Download command to BSIM, slave address=20h (This command instructs the BMC Simulator (BSIM) to send a series of program commands resulting in the download to the UUT of firmware stored in the BSIM’s memory.)
- 4 – Exit F/W Update Mode
- 5 – N/A

This series of commands, when executed in the order they are listed, will transfer a f/w hex file from the BMC Simulator (BSIM) to the UUT.

Command 1 transfers control of the I²C interface from the main interrupt driven program to a polled I²C f/w load program located in memory block 2 of the P89C662.

Command 2 erases memory blocks 0 and 1 where the main program normally resides.

Command 3 instructs the BSIM to send a series of program commands resulting in the download to memory blocks 1 and 2 of the UUT of firmware stored in the BSIM’s memory.

Command 4 causes control to be transferred to the newly downloaded program so initialization can occur and the main loop can begin execution. The UUT will perform normally after this point using the newly updated firmware